Q Write a Program to simulate the working of circularqueue of integers using an array. Provide the following operations

a) Insert

b) Delete

c) Display

The program should print appropriate messages for queue empty and queue overflow conditions

```
#include <stdio.h>
# define N 5

int q [N];
int front = -1, rear = -1;
void insert (int) ;
int delete ();
void display ();
void main()
{
    int n, choice;
    do
    {
        printf ("\n 1. Insert \n 2. Delete \n 3. Display \n 4. Exit \n");
        printf ("Enter your option: \n");
        scanf ("%d", &choice);
        switch (choice)
        {
            case 1:
                printf ("Enter the number to Be inserted in the queue: \n");
                scanf ("%d", &n);
                insert (n);
break;
case 2 :
    n= delete( );
    if (n!=-1)
        printf ("\n The number deleted is : %d \n", n);
    break;
   case 3 :
      display( );
     break;
   case 4 :
      exit (0);
      break;
   default :
    printf ("Invalid option \n");
    exit (0);
    break;
  }
 } while (choice != 4);
}
 void insert (int num)
 {
    if (( front ==0 &&rear ==N-1) || rear(==front -1)))
        printf ("\n Overflow");
     else if ( front == -1 && rear == -1)
     {
          front = rear = 0;
          q(rear ) = num;
     }
     else
     {
          rear ++ ;
```

1/1/24

```
        q[rear] = num;
    }
}
int delete()
{
    int val;
    if (front == -1 && rear == -1)
    {
        printf("\n Underflow");
        return -1;
    }
    val = q[front];
    if (front == rear)
        front = rear = -1;
    else
    {
        if (front == N-1)
            front = 0;
        else
            front ++;
    }
    return val;
}
void display()
{
    int i;
    printf("\n");
    if (front == -1 && rear == -1)
        printf("\n Queue is empty");
    else
    {
```

```
if (front <= rear)
 {
    for (i = front ; i <= rear ; i++)
        printf ("\t %d", q[i]);
   }
   else
   {
      for (i= front ; i< N ; i++ )
         printf ( "\t %d", q[i]);
         for (i=0 ; i<=rear ; i++)
           printf ("\t %d", q[i]));
    }
  }
}
```

Output

```
1. Insert
2 Delete
3. Display
4. Exit
  Enter your option : 1
  Enter the number : 6
```

```
1. Insert
2. Delete
3. Display
4. Exit
 Enter your option : 3
         6
```

```
1. Insert
2. Delete
3. Display
4. Exit
 Enter your option : 2
   The number deleted is : 6
```

```
1. Insert
2. Delete
3. Display
4. Exit
   Enter your option :
       4
```